

VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT
AUF DEM GEBIET DES PATENTWESENS

PCT

INTERNATIONALER RECHERCHENBERICHT

(Artikel 18 sowie Regeln 43 und 44 PCT)

Aktenzeichen des Anmelders oder Anwalts	WEITERES VORGEHEN	siehe Mitteilung über die Übermittlung des internationalen Recherchenberichts (Formblatt PCT/ISA/220) sowie, soweit zutreffend, nachstehender Punkt 5
Internationales Aktenzeichen	Internationales Anmeldedatum (Tag/Monat/Jahr)	(Frühestes) Prioritätsdatum (Tag/Monat/Jahr)
PCT/DE 99/ 00213	28/01/1999	31/01/1998
Anmelder		
ECK, Peter et al.		

Dieser internationale Recherchenbericht wurde von der Internationalen Recherchenbehörde erstellt und wird dem Anmelder gemäß Artikel 18 übermittelt. Eine Kopie wird dem Internationalen Büro übermittelt.

Dieser internationale Recherchenbericht umfaßt insgesamt 2 Blätter.

☒ Darüber hinaus liegt ihm jeweils eine Kopie der in diesem Bericht genannten Unterlagen zum Stand der Technik bei.

1. Grundlage des Berichts

a. Hinsichtlich der **Sprache** ist die internationale Recherche auf der Grundlage der internationalen Anmeldung in der Sprache durchgeführt worden, in der sie eingereicht wurde, sofern unter diesem Punkt nichts anderes angegeben ist.

☐ Die internationale Recherche ist auf der Grundlage einer bei der Behörde eingereichten Übersetzung der internationalen Anmeldung (Regel 23.1 b)) durchgeführt worden.

b. Hinsichtlich der in der internationalen Anmeldung offenbarten **Nucleotid- und/oder Aminosäuresequenz** ist die internationale Recherche auf der Grundlage des Sequenzprotokolls durchgeführt worden, das

☐ in der internationalen Anmeldung in Schriftlicher Form enthalten ist.

☐ zusammen mit der internationalen Anmeldung in computerlesbarer Form eingereicht worden ist.

☐ bei der Behörde nachträglich in schriftlicher Form eingereicht worden ist.

☐ bei der Behörde nachträglich in computerlesbarer Form eingereicht worden ist.

☐ Die Erklärung, daß das nachträglich eingereichte schriftliche Sequenzprotokoll nicht über den Offenbarungsgehalt der internationalen Anmeldung im Anmeldezeitpunkt hinausgeht, wurde vorgelegt.

☐ Die Erklärung, daß die in computerlesbarer Form erfaßten Informationen dem schriftlichen Sequenzprotokoll entsprechen, wurde vorgelegt.

2. ☐ Bestimmte Ansprüche haben sich als nicht recherchierbar erwiesen (siehe Feld I).

3. ☐ Mangelnde Einheitlichkeit der Erfindung (siehe Feld II).

4. Hinsichtlich der Bezeichnung der Erfindung

☒ wird der vom Anmelder eingereichte Wortlaut genehmigt.

☐ wurde der Wortlaut von der Behörde wie folgt festgesetzt:

5. Hinsichtlich der Zusammenfassung

☒ wird der vom Anmelder eingereichte Wortlaut genehmigt.

☐ wurde der Wortlaut nach Regel 38.2b) in der in Feld III angegebenen Fassung von der Behörde festgesetzt. Der Anmelder kann der Behörde innerhalb eines Monats nach dem Datum der Absendung dieses internationalen Recherchenberichts eine Stellungnahme vorlegen.

6. Folgende Abbildung der **Zeichnungen** ist mit der Zusammenfassung zu veröffentlichen: Abb. Nr. 1

☒ wie vom Anmelder vorgeschlagen

☐ keine der Abb.

☐ weil der Anmelder selbst keine Abbildung vorgeschlagen hat.

☐ weil diese Abbildung die Erfindung besser kennzeichnet.

THIS PAGE BLANK (USPTO)

INTERNATIONAL SEARCH REPORT

International Application No.

PCT/DE 99/00213

A. CLASSIFICATION OF SUBJECT MATTER

IPC 6 H03M7/30 H03M7/40 G06F9/45

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 6 H03M G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	R. CAMERON: "Source encoding using syntactic information source models" IEEE TRANSACTIONS ON INFORMATION THEORY, vol. 34, no. 4, July 1988 (1988-07), pages 843-850. XP000212382	1,2
A	paragraph '00II! - paragraph '0III! ---	8
X	J. KATAJAINEN: "Syntax-directed compression of program files" SOFTWARE-PRACTICE AND EXPERIENCE, vol. 16, no. 3, March 1986 (1986-03), pages 269-276. XP002013872 *paragraph "General description"* ---	1,2
X	US 5 678 052 A (BRISSE) 14 October 1997 (1997-10-14) column 3, line 45 - column 6, line 7 -----	1

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

Special categories of cited documents

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

21 July 1999

Date of mailing of the international search report

29/07/1999

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040. Tx. 31 651 epo nl.
Fax: (+31-70) 340-3016

Authorized officer

Augarde, E

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/DE 99/00213

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5678052 A	14-10-1997	NONE	

INTERNATIONALER RECHERCHENBERICHT

Internationales Aktenzeichen

PCT/DE 99/00213

A. KLASSIFIZIERUNG DES ANMELDUNGSGEGENSTANDES
IPK 6 H03M7/30 H03M7/40 G06F9/45

Nach der Internationalen Patentklassifikation (IPK) oder nach der nationalen Klassifikation und der IPK

B. RECHERCHIERTE GEBIETE

Recherchierter Mindestprüfstoff (Klassifikationssystem und Klassifikationssymbole)
IPK 6 H03M G06F

Recherchierte aber nicht zum Mindestprüfstoff gehorende Veröffentlichungen, soweit diese unter die recherchierten Gebiete fallen

Während der internationalen Recherche konsultierte elektronische Datenbank (Name der Datenbank und evtl. verwendete Suchbegriffe)

C. ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
X	R. CAMERON: "Source encoding using syntactic information source models" IEEE TRANSACTIONS ON INFORMATION THEORY, Bd. 34, Nr. 4, Juli 1988 (1988-07), Seiten 843-850, XP000212382	1,2
A	Absatz '00II! - Absatz '0III! ---	8
X	J. KATAJAINEN: "Syntax-directed compression of program files" SOFTWARE-PRACTICE AND EXPERIENCE, Bd. 16, Nr. 3, März 1986 (1986-03), Seiten 269-276, XP002013872 * Absatz "General description"* ---	1,2
X	US 5 678 052 A (BRISSON) 14. Oktober 1997 (1997-10-14) Spalte 3, Zeile 45 - Spalte 6, Zeile 7 -----	1



Weitere Veröffentlichungen sind der Fortsetzung von Feld C zu entnehmen



Siehe Anhang Patentfamilie

* Besondere Kategorien von angegebenen Veröffentlichungen :

"A" Veröffentlichung, die den allgemeinen Stand der Technik definiert, aber nicht als besonders bedeutsam anzusehen ist

"E" älteres Dokument, das jedoch erst am oder nach dem internationalen Anmeldedatum veröffentlicht worden ist

"L" Veröffentlichung, die geeignet ist, einen Prioritätsanspruch zweifelhaft erscheinen zu lassen, oder durch die das Veröffentlichungsdatum einer anderen im Recherchenbericht genannten Veröffentlichung belegt werden soll oder die aus einem anderen besonderen Grund angegeben ist (wie ausgeführt)

"O" Veröffentlichung, die sich auf eine mündliche Offenbarung, eine Benutzung, eine Ausstellung oder andere Maßnahmen bezieht

"P" Veröffentlichung, die vor dem internationalen Anmeldedatum, aber nach dem beanspruchten Prioritätsdatum veröffentlicht worden ist

"T" Spätere Veröffentlichung, die nach dem internationalen Anmeldedatum oder dem Prioritätsdatum veröffentlicht worden ist und mit der Anmeldung nicht kollidiert, sondern nur zum Verständnis des der Erfindung zugrundeliegenden Prinzips oder der ihr zugrundeliegenden Theorie angegeben ist

"X" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann allein aufgrund dieser Veröffentlichung nicht als neu oder auf erfinderscher Tätigkeit beruhend betrachtet werden

"Y" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann nicht als auf erfinderscher Tätigkeit beruhend betrachtet werden, wenn die Veröffentlichung mit einer oder mehreren anderen Veröffentlichungen dieser Kategorie in Verbindung gebracht wird und diese Verbindung für einen Fachmann naheliegend ist

"Z" Veröffentlichung, die Mitglied derselben Patentfamilie ist

Datum des Abschlusses der internationalen Recherche

21. Juli 1999

Absenddatum des internationalen Recherchenberichts

29/07/1999

Name und Postanschrift der Internationalen Recherchenbehörde
Europäisches Patentamt, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Bevollmächtigter Bediensteter

Augarde, E

INTERNATIONALER RESEARCHERBERICHT

Angaben zu Veröffentlichungen, die zur selben Patentfamilie gehören

Internationales Aktenzeichen

PCT/DE 99/00213

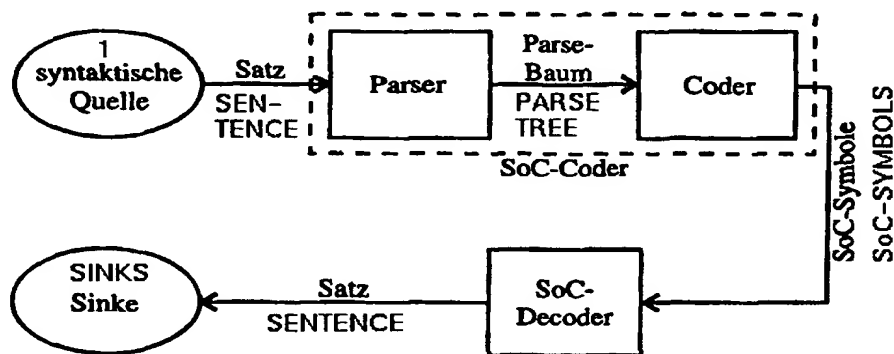
Im Recherchenbericht angeführtes Patentdokument	Datum der Veröffentlichung	Mitglied(er) der Patentfamilie	Datum der Veröffentlichung
US 5678052 A	14-10-1997	KEINE	

INTERNATIONALE ANMELDUNG VERÖFFENTLICHT NACH DEM VERTRAG ÜBER DIE
INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT)

<p>(51) Internationale Patentklassifikation 6 H03M 7/00</p>	<p>A2</p>	<p>(11) Internationale Veröffentlichungsnummer: WO 99/39441</p> <p>(43) Internationales Veröffentlichungsdatum: 5. August 1999 (05.08.99)</p>
<p>(21) Internationales Aktenzeichen: PCT/DE99/00213</p> <p>(22) Internationales Anmeldedatum: 28. Januar 1999 (28.01.99)</p> <p>(30) Prioritätsdaten: 198 03 845.3 31. Januar 1998 (31.01.98) DE</p> <p>(71)(72) Anmelder und Erfinder: ECK, Peter [DE/DE]; Wangenerstrasse 75, D-82319 Starnberg (DE). MATZNER, Rolf [DE/DE]; Georgenstrasse 105, D-80798 München (DE). XIE, Changsong [CN/DE]; Therese-Giehse-Allee 93, D-81739 München (DE).</p>		<p>(81) Bestimmungsstaaten: JP, US, europäisches Patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).</p> <p>Veröffentlicht <i>Ohne internationalen Recherchenbericht und erneut zu veröffentlichen nach Erhalt des Berichts.</i></p>

(54) Title: COMPRESSION OF DATA WITH A SYNTACTIC STRUCTURE

(54) Bezeichnung: KOMPRESSION VON DATEN MIT SYNTAKTISCHER STRUKTUR



1...SYNTACTIC SOURCE

(57) Abstract

The invention relates to the execution of a method and devices which enable the elimination of redundancy in digitally represented messages with a syntactic structure. The data reduction thus results by parsing the message corresponding to the rules of the underlying syntax and to the subsequent entropy coding of the decisions which affect selection of the rules. In the case when the message is a computer program, the original message or directly one executable program can be generated from the compressed representation by means of a pushdown automation. The area of application of the invention is, among others, in the transmission of computer programs in networks having limited bandwidths.

(57) Zusammenfassung

Es wird ein Verfahren und Einrichtungen zu seinem Vollzug vorgeschlagen, die es ermöglichen, Redundanz in digital dargestellten Nachrichten mit syntaktischer Struktur zu eliminieren. Die Datenreduktion erfolgt dabei durch Parsen der Nachricht entsprechend den Regeln der zugrundeliegenden Syntax und nachfolgender Entropiecodierung der zur Auswahl der Regeln getroffenen Entscheidungen. Falls die Nachricht ein Computerprogramm ist, kann aus der komprimierten Darstellung durch einen Kellerautomaten die Originalnachricht oder direkt ein ausführbares Programm erzeugt werden. Anwendungsgebiet ist u.a. die Übertragung von Computerprogrammen in Netzwerken mit begrenzter Bandbreite.

LEDIGLICH ZUR INFORMATION

Codes zur Identifizierung von PCT-Vertragsstaaten auf den Kopfbögen der Schriften, die internationale Anmeldungen gemäss dem PCT veröffentlichen.

AL	Albanien	ES	Spanien	LS	Lesotho	SI	Slowenien
AM	Armenien	FI	Finnland	LT	Litauen	SK	Slowakei
AT	Österreich	FR	Frankreich	LU	Luxemburg	SN	Senegal
AU	Australien	GA	Gabun	LV	Lettland	SZ	Swasiland
AZ	Aserbaidshjan	GB	Vereinigtes Königreich	MC	Monaco	TD	Tschad
BA	Bosnien-Herzegowina	GE	Georgien	MD	Republik Moldau	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagaskar	TJ	Tadschikistan
BE	Belgien	GN	Guinea	MK	Die ehemalige jugoslawische Republik Mazedonien	TM	Turkmenistan
BF	Burkina Faso	GR	Griechenland	ML	Mali	TR	Türkei
BG	Bulgarien	HU	Ungarn	MN	Mongolei	TT	Trinidad und Tobago
BJ	Benin	IE	Irland	MR	Mauretanien	UA	Ukraine
BR	Brasilien	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Island	MX	Mexiko	US	Vereinigte Staaten von Amerika
CA	Kanada	IT	Italien	NE	Niger	UZ	Usbekistan
CF	Zentralafrikanische Republik	JP	Japan	NL	Niederlande	VN	Vietnam
CG	Kongo	KE	Kenia	NO	Norwegen	YU	Jugoslawien
CH	Schweiz	KG	Kirgisistan	NZ	Neuseeland	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Demokratische Volksrepublik Korea	PL	Polen		
CM	Kamerun	KR	Republik Korea	PT	Portugal		
CN	China	KZ	Kasachstan	RO	Rumänien		
CU	Kuba	LC	St. Lucia	RU	Russische Föderation		
CZ	Tschechische Republik	LI	Liechtenstein	SD	Sudan		
DE	Deutschland	LK	Sri Lanka	SE	Schweden		
DK	Dänemark	LR	Liberia	SG	Singapur		
EE	Estland						

Kompression von Daten mit syntaktischer Struktur

1 Einführung in den Themenkreis

Wichtige technische Anwendungen der Quellencodierung liegen auf dem Gebiet der Übertragung und Speicherung von Nachrichten. Aus dem Wunsch einer geringen Datenrate bei der Übertragung von Nachrichten, bzw. eines geringen Platzbedarfs bei der Speicherung von Nachrichten, erwächst die Forderung nach einer möglichst kurzen, d.h. redundanzarmen und im Idealfall redundanzfreien Codierung der Nachricht.

2 Stand der Technik

2.1 Formen von Redundanz

Redundanz tritt in Quellensymbolströmen in zweierlei grundlegenden Formen in Erscheinung: Als unterschiedliche Auftretishäufigkeit der Elemente des Quellensymbolvorrats, und in Form von statistischen Abhängigkeiten zwischen zeitlich auseinanderliegenden Quellensymbolen, die ihre Ursache meist in quellenspezifischen *Bildungsgesetzen* für die Konstruktion von Nachrichten als Folge von Quellensymbolen haben.

3 Nachteile bekannter Verfahren

Zur Kompaktierung redundanzbehafteter Quellen, deren abgegeben Quellensymbole nicht gleichverteilt, aber statistisch unabhängig sind, ist ein Verfahren nach Huffman [1] bekannt. Speziell für diesen Quellentyp geeignet, ist es mit dem Verfahren nur unter der günstigsten Voraussetzung möglich, daß die statistischen Eigenschaften der Quelle bekannt sind oder zuverlässig geschätzt werden können, eine maximale Kompaktierung zu erreichen.

Während die Schätzung der für Huffman-Codierung notwendigen Auftretiswahrscheinlichkeiten einzelner Quellensymbole auch bei mäßig langen Nachrichten ausreichend genau möglich ist, erfordert die Schätzung statistischer Abhängigkeiten zwischen zeitlich auseinanderliegenden Quellensymbolen in der Regel eine umfassende Beobachtung der Quelle, die technische Machbarkeit (Speicherplatz) und verfügbaren Beobachtungszeitraum bei weitem übersteigen. Deswegen treffen das bekannte oder ähnliche Verfahren stets Annahmen über das zugrundeliegende Bildungsgesetz oder wesentliche Charakteristika desselben, und sind dann natürlich auch nur für Quellen mit diesem Bildungsgesetz geeignet. Diese Einschränkung gilt auch für das seit Ende der 70er Jahre bekannte Verfahren nach Lempel-Ziv (LZ) Datenkompaktierungsverfahren [2]. Dieses Verfahren ist zur Kompaktierung beliebiger linearer Symbolfolgen geeignet, d.h. es ist anwendbar ohne Kenntnis der statistischen Eigenschaften der Nachrichtenquelle. Hauptanwendungsgebiete sind heute die Kompaktierung von Bilddaten (GIF-Dateiformat) und beliebiger Dateien (Archivierungsprogramme gzip, pkzip). Im Gegensatz zur Huffman-Codierung

nutzt dieses Verfahren, zwischen zeitlich aufeinanderfolgenden Symbolen bestehende, statistische Abhängigkeiten durch Berücksichtigung wiederholt auftretender Teilsymbolfolgen (Strings) aus. Damit ist LZ-Kompaktierung nur für solche Quellen geeignet, deren Bildungsgesetz zum häufigen Auftreten bestimmter Strings führt. Die Redundanz
40 die durch das Befolgen grammatikalischer Regeln bei der Konstruktion einer Nachricht in diese eingebracht wird, ist den bekannten Kompaktierungsverfahren aufgrund der rekursiven Struktur grammatikalischer Regeln nicht zugänglich, da sie lediglich Symbolhäufigkeiten oder sich wiederholende Zeichenketten zur Kompaktierung heranziehen.

4 Definition der Erfindungsaufgabe

- 45 Aufgabe der Erfindung ist es, ein Verfahren und Einrichtungen zum Vollzug des Verfahrens zu schaffen, welche sich auf die Fähigkeiten erstrecken, aus einer als Zeichenstrom vorliegenden Nachricht, die gemäß grammatikalischen Regeln gebildet wurde, durch Codieren jene Redundanz zu entfernen, die durch die Einschränkung aller möglichen Zeichenfolgen durch die Grammatik entsteht.
- 50 Diese Aufgabe wird durch ein Verfahren mit dem in Anspruch 1 angegebenen Merkmalen und zwei Vorrichtungen, mit den in den Ansprüchen 8 und 9 genannten Merkmalen bewältigt.

5 Vorteile der erfinderischen Lösung

- Mit der erfindungsgemäßen Lösung, eine Datenreduktion in Abhängigkeit zur verwendeten Programmiersprachensyntax vornehmen zu können, ist es nunmehr in besonders
55 vorteilhafter Weise möglich, Redundanz in Nachrichten, die sich während ihrer Konstruktion, wegen der Notwendigkeit bestimmte grammatikalische Regeln beachten zu müssen, asymptotisch nahezu vollständig zu entfernen.

Die Erfindung hat dies durch analytische Berechnung der Kapazität ihres syntax-orientiert codierten Datenstroms am Beispiel einer Sprache mit für Programmiersprachen
60 typischen Charakteristika (mathematische Ausdrücke, if-then-else-Konstrukt, etc.) bewiesen. Dabei zeigte sich, daß ein nicht geringer Anteil der Redundanz, der im Aufgabenbereich herkömmlicher Verfahren (Lempel-Ziv, Huffman) liegt, durch diese bisher nicht beseitigt wird.

- 65 Das neue Verfahren der syntax-orientierten Codierung ist dagegen in der Lage, die Datenmenge z.B. bezogen auf ein Komprimierungsergebnis nach dem Lempel-Ziv-Verfahren, noch einmal um nahezu die Hälfte zu mindern.

Damit ist das erfindungsgemäße Verfahren und zu ihrem Vollzug definierten Einrichtungen für alle Anwendungen prädestiniert, welche die Übertragung oder Speicherung
70 von syntaktisch strukturierten Nachrichten beinhalten. Beispiele sind die Übertragung von Java-Applets im Internet oder Intranet, die Übertragung oder Speicherung von

Postscript-Dateien, die Übertragung und Speicherung von MPEG-4-codierten Videodaten, oder die Verwendung höherer Protokolle bei der Kommunikation. Gerade für die Übertragung von Programmen, wie z. B. Java-Applets, ist SoC in besonderem Maße
 75 geeignet, da der Parser sowohl Bestandteil eines Compilers als auch eines SoC-Coders ist, und somit SoC organisch in den Datenfluß von der Programmerstellung bis zum Programmablauf integriert werden kann.

6 Wirkungsweise der Erfindung

6.1 Syntaktisch strukturierte Quelle

80 Die Erfindung geht von einer formalen Sprache aus, welche durch eine Menge von Zeichenketten dargestellt wird. Die Zusammenstellung der Nachricht erfolgt dabei nach einer der Sprache zugeordneten Grammatik. Eine Grammatik ist sowohl das mathematische System zur Definition einer formalen Sprache, als auch ein Satz von Regeln zur Feststellung der syntaktischen Gültigkeit eines konkreten Satzes. Im folgenden werden
 85 nur kontextfreie Grammatiken betrachtet, da höhere Programmiersprachen (C, C++, Java, etc.) fast ausschließlich durch kontextfreie Grammatiken beschrieben werden. folgendermaßen definiert:

Definition 1 Eine kontextfreie Grammatik ist ein 4-Tupel $G = (N, T, P, S)$, wobei

N die Menge der Nonterminalsymbole,
 90 T die Menge der Terminalsymbole, und
 P die Relation $N \times (N \cup T)^*$ ist.
 S ist ein besonderes Nonterminalsymbol, auch Startsymbol genannt.

Definition 2 Ein Element $(A, \beta) \in P$ heißt Produktion und wird abkürzend $A \rightarrow \beta$ geschrieben.

95 Eine Produktion (oder Ableitung) ist somit eine Ersetzungsregel für ein Nonterminalsymbol A , wobei dieses Nonterminalsymbol durch einen String (Kette) β aus (Nonterminal- und) Terminalsymbolen ersetzt wird.

Existieren mehrere Produktionen $(A, \beta_1), (A, \beta_2), \dots, (A, \beta_n) \in P$, so schreiben wir hierfür:

$$\begin{array}{c} A \rightarrow \beta_1 \\ \quad | \beta_2 \\ \quad \vdots \\ \quad | \beta_n \end{array}$$

Definition 3 Die Menge $P_{A_0} \subseteq P$ aller möglichen Ableitungen für ein Nonterminalsymbol A_0

$$P_{A_0} = \{(A, \beta) \in P : A = A_0\}$$

heißt auch die Menge der Alternativen für A_0 . Die Entscheidung für die Anwendung einer konkreten Produktion $(A_0, \beta_k) \in P_{A_0}, (k = 1, 2, \dots, n)$ nennen wir Auswahl.

Beispiel 1 Alle arithmetischen Ausdrücke in der Variablen a mit den Operationen $+$ und $*$ sowie beliebig verschachtelten Klammern $(,)$ sind eine formale Sprache, die durch die Grammatik

$$G = (\{E, T, F\}, \{a, +, *, (,)\}, P, E)$$

erzeugt wird. P besteht aus den folgenden Produktionen:

$$\begin{array}{lcl} E & \rightarrow & E + T \\ & | & T \\ T & \rightarrow & T * F \\ & | & F \\ F & \rightarrow & (E) \\ & | & a \end{array}$$

100 $a * a + (a * a + a) * a$ ist beispielsweise ein gültiger Satz.

Bei der Erzeugung eines Satzes werden Strings aus Nonterminal- und ggf. Terminalsymbolen durch Anwendung passender Produktionen auf die Nonterminalsymbole weiter abgeleitet, bis letztendlich der String nur noch aus Terminalsymbolen besteht. Im o.g. Beispiel besteht der gültige Satz nur noch aus den Terminalsymbolen $a, +, *, (,)$.

105 6.2 Prinzip der Syntax-orientierten Codierung (SoC)

Information fließt in eine erzeugte Nachricht immer dann ein, wenn bei ihrer Konstruktion Entscheidungen getroffen werden. Entscheidungen sind dann zu treffen, wenn von mehreren möglichen Quellensymbolen eines auszuwählen ist, oder – bei einer syntaktisch strukturierten Quelle – von mehreren anwendbaren Produktionen $A \rightarrow \beta_1, A \rightarrow \beta_2, \dots$
 110 genau eine Produktion Anwendung findet (Auswahl, vgl. Definition 3).

SoC codiert anstelle des Strings aus Terminalsymbolen die Auswahlen, die der Reihe nach bei der Konstruktion dieses Strings (der Nachricht) vorgenommen werden.

Definition 4 $\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_n\}$ heißt SoC-Alphabet mit

$$n = \max_A \{|P_A|\}$$

Alle Elemente einer jeden Menge von Alternativen P_α werden mit SoC-Symbolen $\sigma_i \in \Sigma$ (beliebig) durchnummeriert:

115 **Definition 5** $z_A : P_A \rightarrow \Sigma$ ist eine (beliebige) injektive Abbildung.

Die konkrete Definition von z_A kann i.a. abhängig von P_A sein.

Zur Veranschaulichung der Erfindung sind folgend Ausführungsbeispiele näher beschrieben und in den Figuren 1 bis 6 dargestellt. Es zeigen

- Figur 1 ein Blockschaltbild eines SoC-Übertragungssystems
- 120 • Figur 2 Alternativen der Grammatik G und mit SoC-Symbolen attributierte Auswahlen
- Figur 3 einen Parse-Baum des Satzes $a * a + (a * a + a) * a$ mit Entstehung der linearen SoC-Symbolfolge 121222121121222222
- Figur 1 ein Flußdiagramm der Stack-Maschine (Kellerautomat)
- 125 • Figur 5 ein besonderes Beispiel für ein SoC-Übertragungssystem mit empfängerseitiger Quelltextausgabe
- Figur 5 ein Beispiel für ein SoC-Übertragungssystem mit lauffähigem Java-Applet bzw. -Applikation als Ausgabe

130 Die Funktion eines Übertragungssystems mit SoC ist in Figur 1 veranschaulicht. Grundsätzlich sind Übertragung und Speicherung im Hinblick auf die Wirkungsweise der Codierung gleichzusetzen.

Die Quelle liefert einen syntaktisch strukturierten Symbolstrom (Quellprogramm). Elementarer Bestandteil des SoC-Coders ist der Parser, dessen Aufgabe es ist, aus der zunächst linearen Folge von Quellensymbolen die grammatikalische Struktur des Satzes (des Quellenprogramms) wiederzugewinnen. Diese grammatikalische Struktur wird i.a. 135 durch einen Parse-Baum dargestellt, der seinerseits als Eingabe für den Coder dient. Der Coder erzeugt durch Codierung des Parse-Baumes eine lineare Folge von SoC-Symbolen, wobei das jeweils aktuelle SoC-Symbol in Abhängigkeit der aktuellen Auswahl ausgegeben und übertragen wird. Der SoC-Decoder baut aus der linearen Folge von SoC-Symbolen zunächst den Parse-Baum wieder auf, um dann durch dessen Traversieren die 140 von der Quelle abgegebene Folge von Terminalsymbolen zu rekonstruieren.

6.3 Codierung

Die elementare Aufgabe des Parsers besteht darin, die bei der Konstruktion des Satzes von der Quelle getroffenen *Auswahlen* zu rekonstruieren. Ein Parse-Schritt bedeutet also 145 die Feststellung der konkreten Produktion $(A, \beta) \in P_A$, die auch die Quelle zur Konstruktion der Nachricht an dieser Stelle ausgewählt hatte. Der Coder ordnet entsprechend Definition 5 der solchermaßen festgestellten Auswahl das zugehörige SoC-Symbol $\sigma_k = z_A((A, \beta))$ zu und attributiert den entsprechenden Knoten im Parse-Baum mit σ_k .

Durch Traversieren des Parse-Baumes wird schließlich die lineare Folge von SoC-Symbolen erzeugt und ausgegeben. In den Beispielen dieses Beitrags wird hierfür stets ein Depth-first-Algorithmus verwendet. Der Prozeß der Codierung soll anhand eines beispielhaften Parse-Baumes und der zugrundeliegenden Grammatik aus Beispiel 1 näher erläutert werden.

Beispiel 2 Gegeben ist die kontextfreie Grammatik aus Beispiel 1. Alle Alternativen, d.h. alle Ableitungsmöglichkeiten für jedes Nonterminalsymbol sind in der Figur 2 graphisch dargestellt. Da die mächtigste Menge von Alternativen genau zwei Elemente hat, genügt das SoC-Alphabet $\Sigma = \{1, 2\}$. Jede Auswahl aller Mengen von Alternativen wird mit einem SoC-Symbol attribuiert. Beispielsweise wird die Auswahl der konkreten Produktion $T \rightarrow T * F$ mit dem SoC-Symbol 1 attribuiert.

Ein gültiger Satz der Grammatik ist dann also z. B.: $a * a + (a * a + a) * a$. Den Parse-Baum für diesen Satz und die Entstehung der SoC-Symbolfolge durch Depth-first-Traversieren zeigt Figur 3.

6.4 SoC mit arithmetischer Codierung

Das vorstehend erläuterte Verfahren ist nur dann effizient, wenn

1. $|P_A| = \text{const} \ \forall A$, und
2. $P((A, \beta_i) | A) = \text{const} \ \forall A, \beta_i$,

wobei $P((A, \beta_i) | \alpha)$ die bedingte Wahrscheinlichkeit der β_i erzeugenden Auswahl der Menge der Alternativen P_A bezeichnet.

Diesem Mangel kann durch arithmetische Codierung [3] des SoC-Alphabets Σ gemäß Anspruch 6 und 7 abgeholfen werden. Die für die arithmetische Codierung verwendeten Wahrscheinlichkeitsverteilungen der SoC-Symbole sollten in einer Adaptionseinrichtung nach Anspruch 10 adaptiert werden, wobei die Adaption getrennt für jede Menge von Alternativen durchgeführt werden muß. Anstelle der linearen Sequenz von SoC-Symbolen wird der vom arithmetischen Coder erzeugte Bitstrom übertragen.

6.5 Decodierung

Die Decodierung ist der inverse Prozeß zur Codierung. Der Decoder formatiert die SoC-Codes zurück zum ursprünglichen Satz (Strings von Terminalsymbolen). Er arbeitet nach dem in den Ansprüchen 1, 3 bzw. 4, 5 beschriebenen Verfahren und wird durch eine Stack-Maschine [4] gemäß den Ansprüchen 8 bzw. 9 realisiert.

Ein Flußdiagramm für ein Verfahren nach Anspruch 22 ist in Figur 4 angegeben.

Zur Erläuterung der Stack-Maschine werden folgende Bezeichnungen eingeführt:

β_k String von Terminal- und Nonterminalsymbolen

σ SoC-Symbol

V Terminal- oder Nonterminalsymbol

185 S Startsymbol (Nonterminalsymbol)

Die Decodierung läuft damit in folgenden Schritten ab (die Nummern korrespondieren mit den Bezugszeichen in Figur 4):

1. Die Entwicklung eines gültigen Satzes der Grammatik (eines Programms) beginnt mit dem Startsymbol S . Das Startsymbol S wird auf den Stack gelegt.
- 190 2. Das oberste Symbol V wird vom Stack gelesen.
3. Prüfen, ob V ein Nonterminalsymbol ist.
4. Vom linearen Eingangs-SoC-Symbolstrom wird das nächste SoC-Symbol σ gelesen.
5. Falls V ein Nonterminalsymbol ist, wird die Ableitung mit V weiterentwickelt. Durch σ wird eine Auswahl $(V, \beta_k) = z^{-1}(\sigma)$ der Menge der Alternativen von
195 $V \rightarrow \beta_1 | \beta_2 | \dots | \beta_n$ für V bestimmt. Entsprechend der konkreten Auswahl wird V durch den String β_k ersetzt.
6. Der String β_k von Symbolen (Terminal- und/oder Nonterminalsymbolen) wird auf den Stack gelegt.
7. Falls V ein Terminalsymbol ist, wird es ausgegeben.
- 200 8. Prüfen, ob der Stack leer ist.
9. Die Stack-Maschine terminiert, wenn der Stack leer ist.

Die sukzessive ausgegebenen Strings von Terminalsymbolen ergeben miteinander verkettet schließlich den ursprünglich codierten Satz der Quelle.

7 Ausführungsbeispiel: bevorzugte Form der Anwendung

- 205 Die beiden in Figur 5 und 6 dargestellten Ausführungsbeispiele unterscheiden sich lediglich durch die Datenausgabe am Empfänger (Ausgabe von Quelltext entsprechend dem Verfahren nach Anspruch 3, bzw. direkte Ausgabe eines lauffähigen Java-Applets bzw. einer lauffähigen Java-Applikation entsprechend dem Verfahren nach Anspruch 4).

1. Java-Quelle
210 Die Java-Quelle entspricht einem auf der Java-Grammatik beruhenden ASCII-Text, welcher aus Java-Programmtext sowie Kommentaren, Leerzeichen, usw. besteht.
2. SoC-Encoder
215 Der Soc-Encoder setzt sich zusammen aus einem Compiler-Frontend ((3),(4) und (6)), einer Parse-Baum-Verarbeitung (5), einer Adaptiervorrichtung (8) und Modulen zur Kompression und Verkettung der erzeugten Daten.
3. Scanner
220 Der Scanner liest den Quelltext, entfernt die darin für das Programm unwichtigen Textstellen, wie Dokumentationen, Kommentare, Leerzeichen, usw. und übergibt dem Parser (4) (Terminal-) Symbole.
4. Parser
225 Der Parser interpretiert die (Terminal-) Symbole des Scanners (3), prüft ob die Symbolfolgen mit den grammatikalischen Regeln für Java übereinstimmen und erstellt den für dieses Programm spezifischen Parse-Baum. Zusätzlich wird die Symboltabelle (6) aufgebaut.
5. Parse-Baum-Verarbeitung
230 Bei der Initialisierung der Parse-Baum-Verarbeitung werden alle Knoten des Parse-Baumes mit für jeden Knotentyp festgelegten SoC-Symbolen attribuiert. Die darauf folgende Traversierung aller Knoten des Parse-Baumes führt bei jedem Durchschreiten eines Knotens zur Ausgabe eines SoC-Symbols. Diese SoC-Symbole werden sowohl der Adaptiervorrichtung (8) als auch dem arithmetischen Coder (10) übergeben. Zusätzlich übergibt die Parse-Baum-Verarbeitung den aktuellen Knotentyp an die Adaptiervorrichtung (8).
6. Symboltabellenspeicher
235 Der Symboltabellenspeicher beinhaltet die vom Parser (4) extrahierten Bezeichner des Java-Programmtextes.
7. Zip-Kompressionseinrichtung
Der Inhalt der Symboltabelle (6) wird durch das Verfahren nach Lempel-Zip-Welch [2] komprimiert.
- 240 8. Sendeseitige Adaptiervorrichtung
Die Adaptiervorrichtung trägt bei der Initialisierung festgelegte Anfangswahrscheinlichkeitsverteilungen für jeden Knotentyp in der Wahrscheinlichkeitsverteilungstabelle (9) ein. Bei der laufenden Verarbeitung der von der Parse-Baum-Verarbeitung (5) übernommenen SoC-Symbole und anhand der aktuellen Knotentypen wird bei jedem Schritt die Wahrscheinlichkeitsverteilung aller SoC-Symbole
245

des aktuellen Knotentyps adaptiert und in der Wahrscheinlichkeitsverteilungstabelle (9) eingetragen. Die Adaptiervorrichtung stellt gleichzeitig dem arithmetischen Coder 10 die zum jeweils aktuellen SoC-Symbol gehörende Wahrscheinlichkeitsverteilung zur Verfügung.

250 9. Tabelle der Wahrscheinlichkeitsverteilungen der SoC-Symbole

Für jeden Knotentyp enthält die Tabelle eine eigene Wahrscheinlichkeitsverteilung der SoC-Symbole des jeweiligen Knotentyps.

10. Arithmetischer Coder

255 Der arithmetische Coder [3] empfängt von der Parse-Baum-Verarbeitung (5) das nächste zu codierende SoC-Symbol und codiert es unter Berücksichtigung der von der Adaptiervorrichtung (8) zur Verfügung gestellten Wahrscheinlichkeitsverteilung. Der Output des arithmetischen Coders ist ein binärer Datenstrom.

11. Datenverkettung

260 Der binäre Datenstrom des arithmetischen Coders (10) und die durch (7) komprimierte Symboltabelle (6) werden zu einem Datenpaket zusammengefaßt.

12. HTTP-Server

Auf dem HTTP-Server wird das in der Datenverkettung (11) erstellte Datenpaket abgelegt und zum Abruf auf einem Massenspeicher zur Verfügung gestellt.

13. Intranet/Internet

265 14. HTTP-Client (Internet-Browser)

Der Browser hat die Aufgabe das Datenpaket über das Intranet/Internet 13 von einem HTTP-Server (12) zu laden und den Decodiervorgang durch den SoC-Decoder (15) anzustoßen.

15. SoC-Decoder

270 Der SoC-Decoder besteht im wesentlichen aus dem arithmetischen Decoder (17) und dem Kellerautomaten (22). Als Ergebnis liefert er die Sequenz der (Terminal-) Symbole die dem ursprünglichen Java-Quellcode entsprechen (vgl. 3 und 4). (1).

16. Datenextraktion

275 Hier werden der binäre Datenstrom des arithmetischen Coders (10) und die durch Zip (7) komprimierte Symboltabelle aus dem gemeinsamen Datenpaket extrahiert und ersterer an den arithmetischen Decoder (17), letzterer an die Zip-Dekompressionseinrichtung (18) weitergegeben.

17. Arithmetischer Decoder

280 Der arithmetische Decoder [3] decodiert adaptiv den binären Datenstrom von der Datenextraktion (16) unter Berücksichtigung der von der empfangsseitigen Adaptiervorrichtung (20) zur Verfügung gestellten Wahrscheinlichkeitsverteilungen zu

den SoC-Symbolen und übergibt pro Decodierschritt ein SoC-Symbol an den Kellerautomaten (22).

18. Zip-Dekompressionseinrichtung

285 Die mit dem Verfahren nach Lempel-Ziv (7) komprimierte Symboltabelle des Coders (6) aus der Datenextraktion (16) wird entpackt und an den Symboltabelleispeicher des Decoders (19) eingetragen.

19. Symboltabelleispeicher des Decoders

20. Adaptiervorrichtung (empfangsseitig)

290 Die Initialisierung der empfangsseitigen Adaptiervorrichtung erfolgt analog zur sendeseitigen Adaptiervorrichtung (8). Während der laufenden Decodierung erhält die Adaptiervorrichtung bei jedem Decodierschritt ein SoC-Symbol, adaptiert die Wahrscheinlichkeitsverteilung der SoC-Symbole des aktuellen Knotentyps und trägt die adaptierten Wahrscheinlichkeitsverteilungen in die Wahrscheinlichkeitsverteilungstabelle (21) ein. Gleichzeitig stellt die empfangsseitige Adaptiervorrichtung die für den nächsten Decodierschritt gültige Wahrscheinlichkeitsverteilung dem arithmetischen Decoder zur Verfügung. Die Auswahl dieser Verteilung bestimmt sich durch den vom Kellerautomaten (22) festgelegten und im nächsten Decodierschritt zu erwartenden Knotentyp.

300 21. Tabelle der Wahrscheinlichkeitsverteilungen der SoC-Symbole

Für jeden Knotentyp enthält die Tabelle eine eigene Wahrscheinlichkeitsverteilung der SoC-Symbole dieses Knotentyps.

22. Kellerautomat

305 Die Initialisierung des Kellerautomaten, die Bearbeitung des Stapelspeichers, sowie die Ausgabe von Terminalsymbolen ist in Figur 4 als Flußdiagramm dargestellt. Zusätzlich übergibt der Kellerautomat Daten an die Adaptiervorrichtung(20):

- (a) den jeweils aktuellen Knotentyp zur Adaption der Wahrscheinlichkeitsverteilungen der SoC-Symbole dieses Knotentyps
 - (b) den nächsten Knotentyp, dessen Wahrscheinlichkeitsverteilung der SoC-Symbole die Adaptiervorrichtung (20) im nächsten Decodierschritt dem arithmetischen Decoder (17) zur Verfügung stellen muß.
- 310

23. Stapelspeicher

Stapelspeicher (Stack) für den Kellerautomaten (22).

24. Java-Sinke

315 Wiedergewonnener Java-Programmtext ohne Kommentare. überflüssige Leerzeichen, usw. .

25. Codeerzeuger

Der Codeerzeuger entspricht einem Compiler-Backend und generiert Bytecode für die JVM (26) in Abhängigkeit der Sequenz der (Terminal-) Symbole, die der Kellerautomat (22) liefert.

26. JVM

Die virtuelle Maschine für Java-Programme (Java Virtual Machine).

Literatur

- [1] HUFFMAN, D. A.: *A Method for the Construction of Minimum-Redundancy Codes*. Proc. IRE, 40:1098–1101, 1952.
- [2] ZIV, JACOB und ABRAHAM LEMPEL: *A Universal Algorithm for Sequential Data Compression*. IEEE Trans. Inform. Theory, IT-23(3):337–343, Mai 1977.
- [3] WITTEN, IAN H., RADFORD M. NEAL und JOHN G. CLEARY: *Arithmetic Coding for Data Compression*. Commun. ACM, 30(6):520–540, Juni 1987.
- [4] AHO, ALFRED V., RAVI SETHI und JEFFREY D. ULLMAN: *Compilerbau*. Addison-Wesley Publishing Company, Bonn, 4. Auflage, 1988.

Bezugszeichenliste**Bezugszeichen zur Figur 4**

- β_k String von Terminal- und Nonterminalsymbolen
- σ SoC-Symbol
- V Terminal- oder Nonterminalsymbol
- S Startsymbol (Nonterminalsymbol)
- 1. Die Entwicklung eines gültigen Satzes der Grammatik (eines Programms) beginnt mit dem Startsymbol S . Das Startsymbol S wird auf den Stack gelegt.
- 2. Das oberste Symbol V wird vom Stack gelesen.
- 3. Prüfen, ob V ein Nonterminalsymbol ist.
- 4. Vom linearen Eingangs-SoC-Symbolstrom wird das nächste SoC-Symbol σ gelesen.
- 5. Falls V ein Nonterminalsymbol ist, wird die Ableitung mit V weiterentwickelt. Durch σ wird eine Auswahl $(V, \beta_k) = z^{-1}(\sigma)$ der Menge der Alternativen von $V \rightarrow \beta_1 | \beta_2 | \dots | \beta_n$ für V bestimmt. Entsprechend der konkreten Auswahl wird V durch den String β_k ersetzt.
- 6. Der String β_k von Symbolen (Terminal- und/oder Nonterminalsymbolen) wird auf den Stack gelegt.

7. Falls V ein Terminalsymbol ist, wird es ausgegeben.
350 8. Prüfen, ob der Stack leer ist.
9. Die Stack-Maschine terminiert, wenn der Stack leer ist.

Bezugszeichen zu den Figuren 5 und 6

1. Java-Quelle
2. SoC-Encoder
- 355 3. Scanner
4. Parser
5. Parse-Baum-Verarbeitung
6. Symboltabellenspeicher
7. Zip-Kompressionseinrichtung
- 360 8. Sendeseitige Adaptiervorrichtung
9. Tabelle der Wahrscheinlichkeitsverteilungen der SoC-Symbole
10. Arithmetischer Coder
11. Datenverkettung
12. HTTP-Server
- 365 13. Intranet/Internet
14. HTTP-Client (Internet-Browser)
15. SoC-Decoder
16. Datenextraktion
17. Arithmetischer Decoder
- 370 18. Zip-Dekompressionseinrichtung
19. Symboltabellenspeicher des Decoders
20. Adaptiervorrichtung (empfangsseitig)
21. Tabelle der Wahrscheinlichkeitsverteilungen der SoC-Symbole
22. Kellerautomat
- 375 23. Stapelspeicher
24. Java-Sinke
25. Codeerzeuger
26. JVM

Patentansprüche

- 380 1. Verfahren zur komprimierten Übertragung und/oder Speicherung von einer durch digitale Daten repräsentierten Nachricht, deren Struktur durch eine Grammatik definiert wird,
dadurch gekennzeichnet, daß
- 385 (a) vor der Speicherung und/oder Übertragung die Nachricht in eine lineare Sequenz von Symbolen, welche die sukzessive Anwendung grammatikalischer Regeln zur Bildung der Nachricht definiert, überführt wird, und
- (b) nach der Speicherung und/oder Übertragung zur Decodierung für jedes so syntax orientiert codierte (SoC) Symbol die dem jeweiligen SoC-Symbol entsprechende grammatikalische Regel ausgeführt wird und durch diese Regel
390 bestimmte Ausgangsdaten erzeugt werden.
2. Verfahren nach Anspruch 1,
dadurch gekennzeichnet, daß zur Erzeugung der SoC-Symbole
- (a) durch Parsen der Nachricht ein den aufeinanderfolgenden Anwendungen der grammatikalischen Regeln entsprechender Parse-Baum erzeugt wird,
- 395 (b) jeder Knoten dieses Parse-Baumes mit einem SoC-Symbol attribuiert wird, welches unter allen an dieser Stelle durch die Grammatik erlaubten Regeln die tatsächlich zur Erzeugung der ursprünglichen Nachricht angewendete Regel eindeutig identifiziert, und
- (c) die SoC-Symbole gemäß einer festgelegten Reihenfolge der Traversierung aller Knoten des Parse-Baumes zu einer linearen Sequenz von SoC-Symbolen
400 verkettet werden.
3. Verfahren nach Anspruch 1,
dadurch gekennzeichnet, daß
zur Decodierung
- 405 (a) der oberste Eintrag eines Stapelspeichers gemäß der durch das eingegebene SoC-Symbol bestimmten Produktion der Grammatik substituiert wird,
- (b) noch nicht vollständig bearbeitbare Teile der grammatikalischen Regel in einem Stapelspeicher ablegt werden, und
- (c) vollständig substituierte Teile der Produktion als Teil der ursprünglichen
410 Nachricht ausgegeben werden.
4. Verfahren nach Anspruch 1,
dadurch gekennzeichnet, daß
zur Decodierung

- 415 (a) der oberste Eintrag eines Stapelspeichers gemäß der durch das eingegebene SoC-Symbol bestimmten Produktion der Grammatik substituiert wird,
- (b) noch nicht vollständig bearbeitbare Teile der grammatikalischen Regel in einem Stapelspeicher ablegt werden.
- 420 (c) vollständig substituierte Teile der Produktion unmittelbar zu einem Teil eines ausführbaren Programms für einen realen Prozessor oder eine virtuelle Maschine verarbeitet werden.

5. Verfahren nach Anspruch 3 oder 4

dadurch gekennzeichnet, daß ein Kellerautomat

- 425 (a) durch das Ablegen eines definierten Startsymbols (Nonterminalsymbol) auf den leeren Stapelspeicher initialisiert wird,
- (b) das oberste Symbol vom Stapelspeicher liest,
- (c) prüft, ob das gelesene Symbol ein Terminal- oder Nonterminalsymbol ist,
- (d) falls es ein Terminalsymbol ist, das Symbol ausgibt und abhängig davon, ob weitere Symbole im Stapelspeicher vorhanden sind, mit 5b fortfährt bzw. terminiert, falls der Stapelspeicher leer ist, oder
- 430 (e) falls es ein Nonterminalsymbol ist, das nächste SoC-Symbol vom Eingabestrom liest,
- (f) abhängig vom gelesenen SoC-Symbol genau eine Alternative (Kette von Terminal- und/oder Nonterminalsymbolen) aus der für das aktuell bearbeitete Nonterminalsymbol gültigen Menge von alternativ anwendbaren Ersetzungs-
- 435 regeln (Produktionen) auswählt und
- (g) diese Kette von Terminal- und/oder Nonterminalsymbolen auf den Stapelspeicher legt und mit 5b fortfährt.

6. Verfahren nach Anspruch 2,

dadurch gekennzeichnet, daß

- 440 (a) jeder Knoten des Parse-Baumes mit einem SoC-Symbol und der Wahrscheinlichkeitsverteilung aller in diesem Knoten möglichen SoC-Symbole attribuiert wird.
- (b) die lineare Sequenz der SoC-Symbole in Einheit mit den zugeordneten Wahrscheinlichkeitsverteilungen einer Entropie-Codierung unterworfen wird,
- 445 (c) die Entropie-Decodierung mit den identischen Wahrscheinlichkeitsverteilungen der SoC-Symbole durchgeführt wird, wie sie bei der Entropie-Codierung angewendet wurden.

7. Verfahren nach Anspruch 6,

dadurch gekennzeichnet, daß

- 450 (a) die Wahrscheinlichkeitsverteilung der in einem Knoten anwendbaren Regeln, ausgehend von einer Anfangsverteilung, bei jedem Auftreten eines SoC-Symbols so adaptiert wird, daß die Wahrscheinlichkeit für das aufgetretene SoC-Symbol erhöht und die Wahrscheinlichkeit aller anderen Symbole entsprechend verringert wird.
- 455 (b) die jeweils aktuelle Verteilung der Auftrittswahrscheinlichkeiten den SoC-Symbolen des entsprechenden Knotentyps zugeordnet wird,
- (c) die Wahrscheinlichkeitsverteilung aller SoC-Symbole im aktuellen Knoten zusammen mit dem zu codierenden SoC-Symbol das Modell für eine arithmetische Codierung bildet,
- 460 (d) bei der Decodierung das Ende der Nachricht dadurch erkannt wird, daß der Stapelspeicher leer ist und
- (e) auf das bei der arithmetischen Codierung notwendige End-Of-Message (EOM) Symbol verzichtet wird.

8. Einrichtung für das Verfahren nach Anspruch 1

- 465 gekennzeichnet durch
einen Codierer, bestehend aus
- (a) einem Scanner zur Umformung der Nachricht als Sequenz von lesbaren Zeichen in eine Folge von Terminalsymbolen,
- (b) einem Parser zum Auffinden der grammatikalischen Regeln, durch deren suk-
- 470 zessive Anwendung die Folge von Terminalsymbolen ursprünglich erzeugt wurde,
- (c) einem Mapper, der den vom Parser identifizierten Regeln eindeutig Syntax-orientierte Symbole zuordnet und diese in festgelegter Reihenfolge ausgibt,
- und einen Decodierer, bestehend aus
- 475 (a) einem Kellerautomaten, der, entsprechend dem obersten Symbol des Stapelspeichers und ggf. dem anliegenden SoC-Symbol, das bereits feststehende Terminalsymbol ausgibt, oder die dem aktuellen Symbol zugeordnete Sequenz von Terminal- und/oder Nonterminalsymbolen auf dem Stapelspeicher ablegt und
- 480 (b) einem Lexikon, das die Terminalsymbole wieder durch Ketten aus lesbaren Zeichen ersetzt.

9. Einrichtung für das Verfahren nach Anspruch 3

- gekennzeichnet durch
einen Codierer, bestehend aus

- 485 (a) einem Scanner zur Umformung eines, im Quelltext oder einer vom Quelltext durch einen Präprozessor abgeleiteten Form vorliegenden, Programms in eine Folge von Terminalsymbolen.
- (b) einem Parser zum Auffinden der grammatikalischen Regeln, durch deren sukzessive Anwendung die Folge von Terminalsymbolen ursprünglich erzeugt wurde.
- 490 (c) einem Mapper, der den vom Parser identifizierten Regeln eindeutig Syntaxorientierte Symbole zuordnet und diese in festgelegter Reihenfolge ausgibt, und einen Decodierer, bestehend aus
- (a) einem Kellerautomaten, der, entsprechend dem obersten Symbol des Stapelspeichers und ggf. dem anliegenden SoC-Symbol, das bereits feststehende Terminalsymbol ausgibt, oder die dem aktuellen Symbol zugeordnete Sequenz von Terminal- und/oder Nonterminalsymbolen auf dem Stapelspeicher ablegt und
- 495 (b) einem Codegenerator, der aus der Folge von Terminalsymbolen ausführbaren Maschinencode, oder Zwischencode zur Ausführung auf einer virtuellen Maschine erzeugt.
- 500

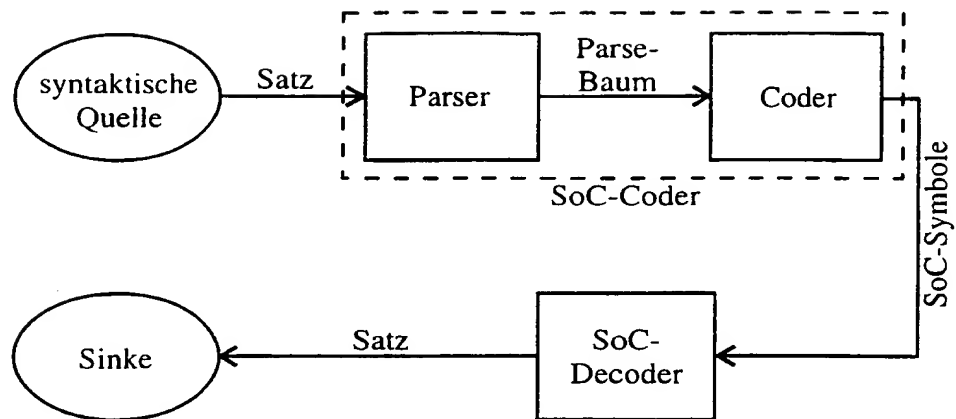
10. Einrichtung für das Verfahren nach Anspruch 7

bestehend aus

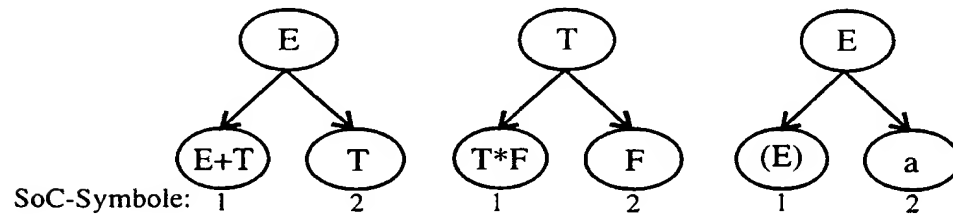
- (a) senderseitig
- 505 i. einer Tabelle, die die Wahrscheinlichkeitsverteilungen der SoC-Symbole für jeden Knotentyp beinhaltet und deren Inhalt bei der Initialisierung mit festgelegten Anfangswahrscheinlichkeitsverteilungen für jeden Knotentyp belegt wird,
- 510 ii. einer Adaptionsvorrichtung, die die Wahrscheinlichkeitsverteilung der SoC-Symbole des momentan gültigen Knotentyps anhand der bestehenden Wahrscheinlichkeitsverteilung, des zu codierenden SoC-Symbols und des aktuellen Knotentyps aktualisiert und diese neue Wahrscheinlichkeitsverteilung in die Tabelle einträgt,
- 515 iii. und einer arithmetischen Codiervorrichtung, die das jeweils zu codierende SoC-Symbol mit der von der Adaptionsvorrichtung zur Verfügung gestellten aktuell gültigen Wahrscheinlichkeitsverteilung codiert;
- (b) und empfängerseitig
- i. einer Tabelle, die die Wahrscheinlichkeitsverteilungen der SoC-Symbole für jeden Knotentyp beinhaltet und deren Inhalt bei der Initialisierung mit festgelegten Anfangswahrscheinlichkeitsverteilungen für jeden Knotentyp belegt wird,
- 520

- 525 ii. einer Adaptionsvorrichtung, die die Wahrscheinlichkeitsverteilung der SoC-Symbole des vom Kellerautomaten festgelegten, momentan gültigen Knotentyps anhand der bestehenden Wahrscheinlichkeitsverteilung, des decodierten SoC-Symbols und des aktuellen Knotentyps aktualisiert und diese neue Wahrscheinlichkeitsverteilung in die Tabelle einträgt,
- 530 iii. einer arithmetischen Decodiervorrichtung, die anhand der von der Adaptionsvorrichtung zur Verfügung gestellten, aktuell gültigen Wahrscheinlichkeitsverteilung des aktuellen Knotentyps das nächste SoC-Symbol decodiert und zur weiteren Verarbeitung dem Kellerautomaten übergibt.

THIS PAGE BLANK (USPTO)



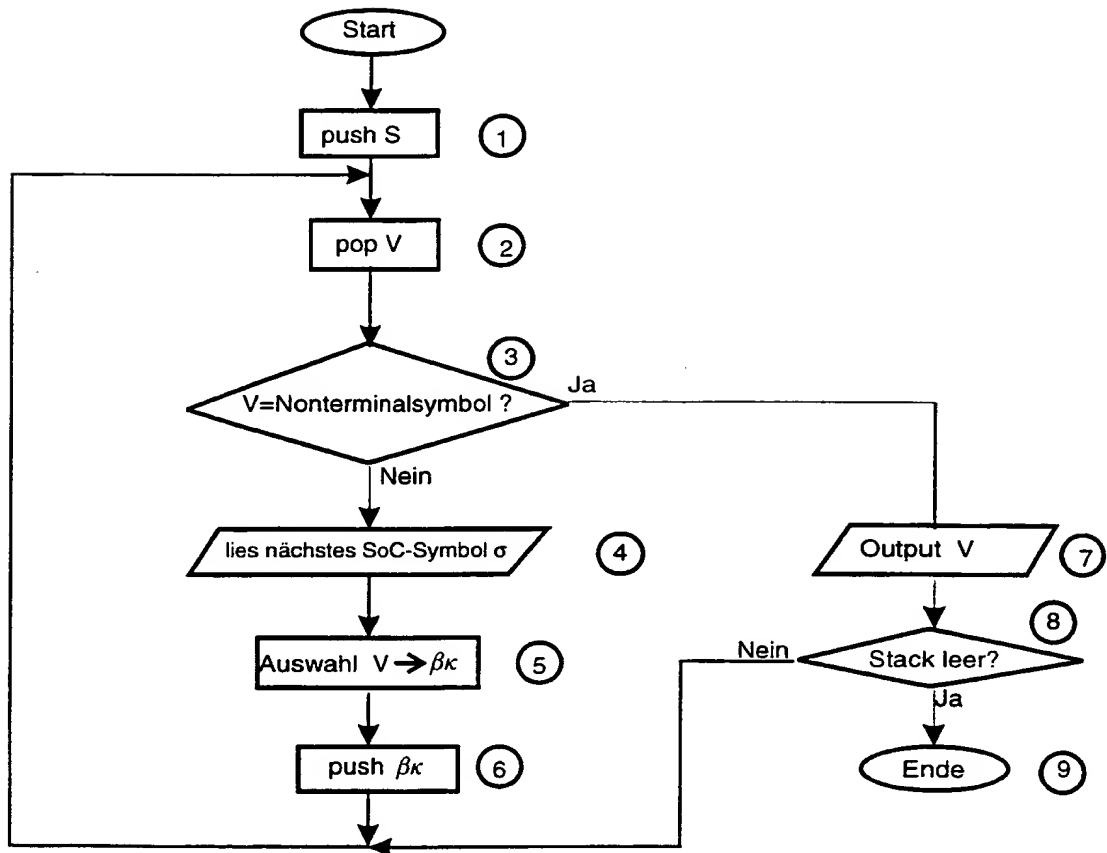
Figur 1



Figur 2

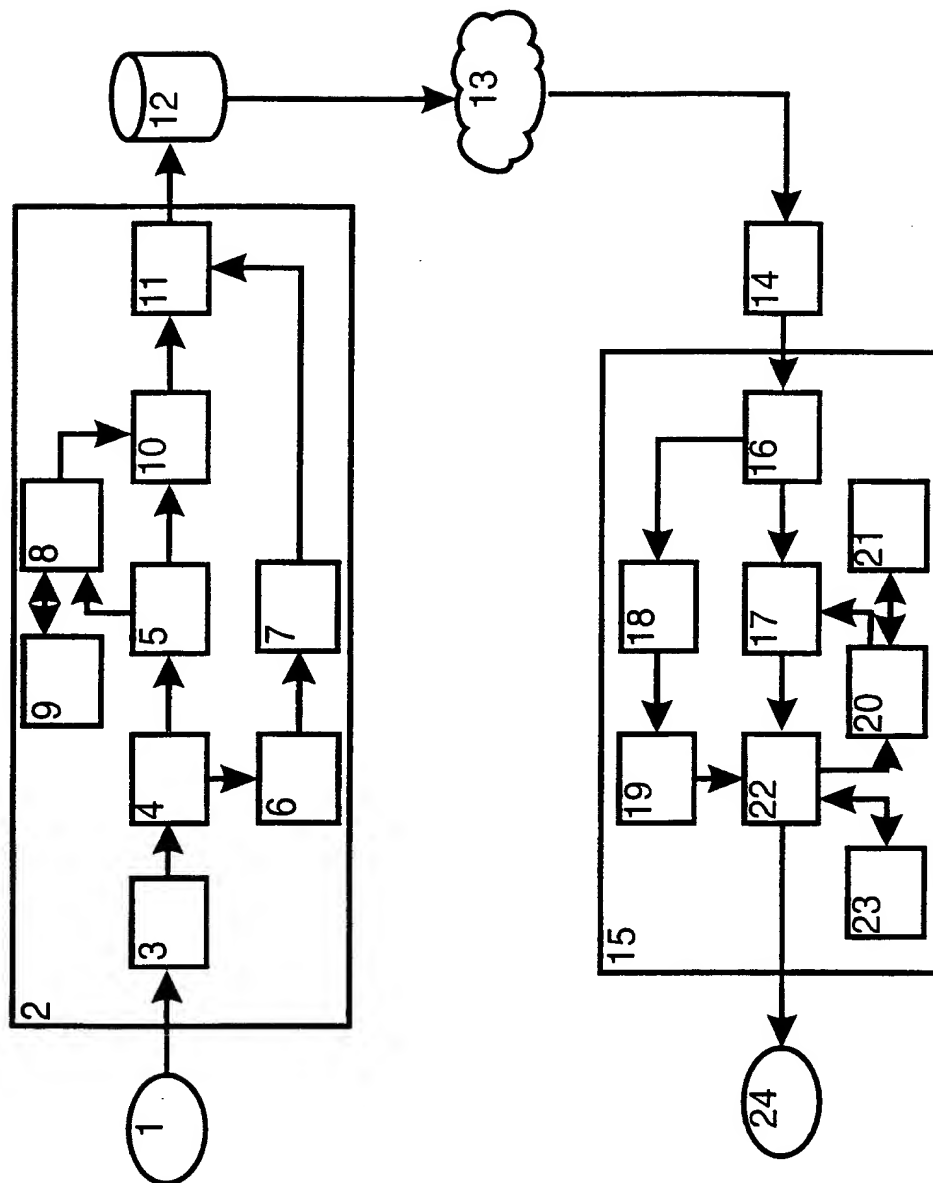
THIS PAGE BLANK (USPTO)

THIS PAGE BLANK (USPTO)



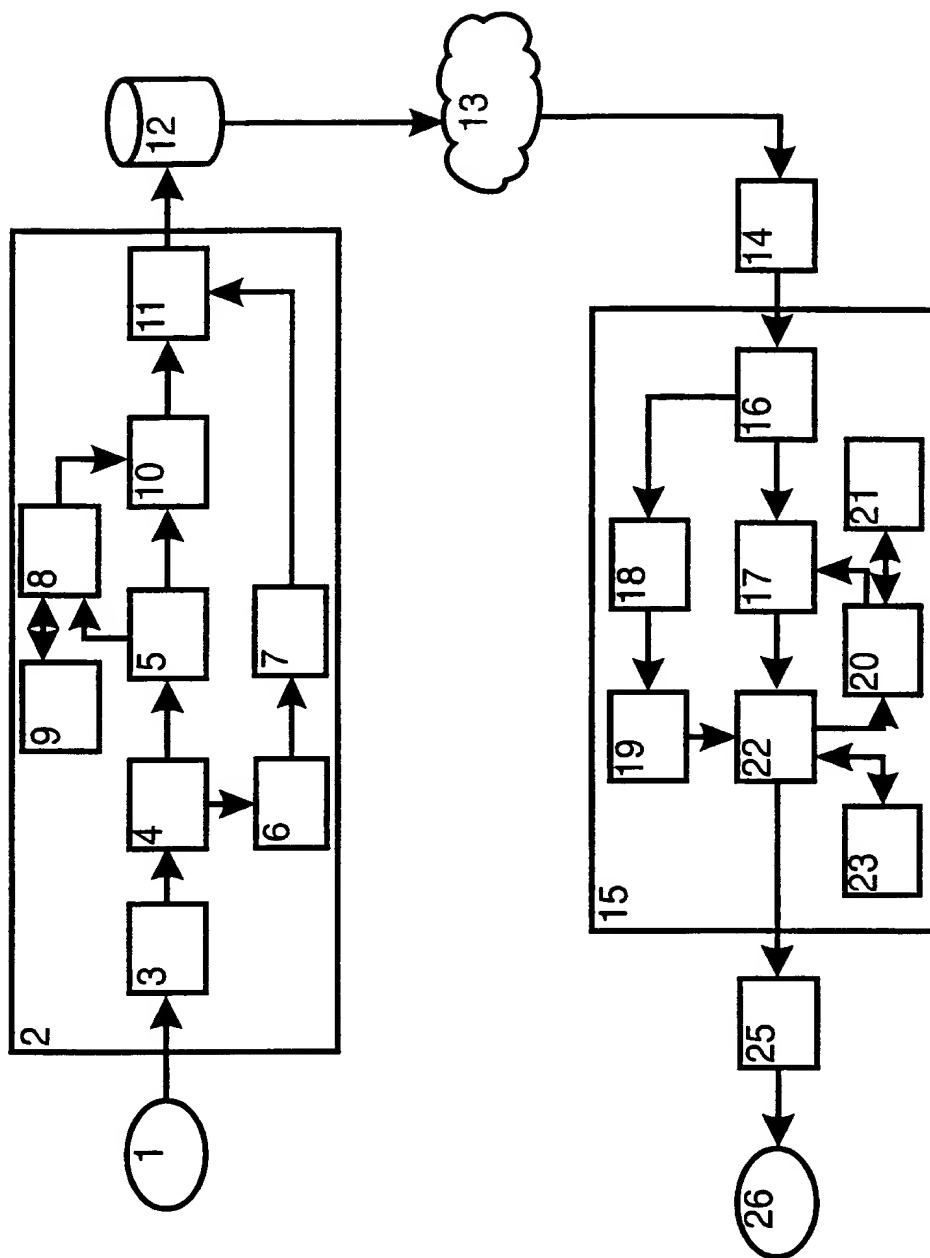
Figur 4

THIS PAGE BLANK (USPTO)



Figur 5

THIS PAGE BLANK (USPTO)



Figur 6

THIS PAGE BLANK (USPTO)

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

THIS PAGE BLANK (USPTO)